

# Soft McEliece: MDPC code-based McEliece cryptosystems with very compact keys through real-valued intentional errors

Marco Baldi, Paolo Santini, Franco Chiaraluce,  
DII, Università Politecnica delle Marche,  
Ancona, Italy

Email: {m.baldi, f.chiaraluce}@univpm.it, S1069914@studenti.univpm.it

**Abstract**—We propose to use real-valued errors instead of classical bit flipping intentional errors in the McEliece cryptosystem based on moderate-density parity-check (MDPC) codes. This allows to exploit the error correcting capability of these codes to the utmost, by using soft-decision iterative decoding algorithms instead of hard-decision bit flipping decoders. However, soft reliability values resulting from the use of real-valued noise can also be exploited by attackers. We devise new attack procedures aimed at this, and compute the relevant work factors and security levels. We show that, for a fixed security level, these new systems achieve the shortest public key sizes ever reached, with a reduction up to 25% with respect to previous proposals.

**Index Terms**—Compact keys, LDPC codes, McEliece cryptosystem, MDPC codes, real-valued intentional errors.

## I. INTRODUCTION

Quantum computers are able to penetrate hard cryptographic targets, like cryptosystems based on integer factorization and discrete logarithms, and code-based cryptosystems are among the most promising solutions able to resist quantum computer-based attacks. The McEliece cryptosystem [1] is the best known code-based asymmetric cryptosystem. In its original formulation based on Goppa codes, it achieves very fast encryption and decryption but has very large public keys, which is a major drawback. According to [2], for achieving 80-bit security with the Goppa code-based cryptosystem we need 460647-bit public keys. A known way to reduce the public key size is to replace Goppa codes with other families of codes, although this may expose the system to security flaws. A recent line of research has been focused on the use of quasi-cyclic low-density parity-check (QC-LDPC) and quasi-cyclic moderate-density parity-check (QC-MDPC) codes in this context, showing that practical systems with compact keys can be designed while preserving the system security [3]–[9].

This has been achieved by keeping the same structure of the original McEliece cryptosystem, in which binary intentional errors are used during encryption. Therefore, hard-decision decoders like the bit flipping iterative decoder [10] are commonly used in these systems. Message-passing decoders can also be used, but without the availability of soft reliability values concerning the ciphertext bits. This makes such decoders work

in suboptimal conditions, that penalize their performance. In fact, low-density parity-check (LDPC) codes achieve the best performance under soft-decision message-passing decoding when soft reliability information is available [11]. For such a reason, in this paper we propose to use real-valued noise samples as an intentional impairment during encryption. This allows to exploit powerful soft-decision decoders to improve the error correcting capability of the legitimate receiver. On the other hand, soft reliability information can be exploited by attackers as well. Therefore, we develop an updated security analysis taking this fact into account. Our results show that this approach allows to achieve public key size reductions up to 25% with respect to the previously known best solutions.

The paper is organized as follows: in Section II we recall the original QC-LDPC and QC-MDPC code-based McEliece cryptosystems, in Section III we introduce a new QC-MDPC code-based variant exploiting real-valued intentional noise, in Section IV we assess security of the new system, in Section V we provide some design examples and in Section VI we draw some conclusive remarks.

## II. QC-LDPC AND QC-MDPC CODE-BASED MCELIECE CRYPTOSYSTEMS

The original QC-LDPC and QC-MDPC code-based McEliece cryptosystems exploit codes having rate  $R = \frac{n_0-1}{n_0}$ , where  $n_0$  is a small integer (e.g.,  $n_0 = 2, 3, 4$ ), redundancy  $r$ , length  $n = n_0 \cdot r$  and dimension  $k = (n_0 - 1) \cdot r$ . The secret code is defined through a sparse parity-check matrix  $\mathbf{H}$  having the following form [4], [12]:

$$\mathbf{H} = [\mathbf{H}_0 | \mathbf{H}_1 | \dots | \mathbf{H}_{n_0-1}], \quad (1)$$

where each block  $\mathbf{H}_i$  is a circulant matrix with size  $r \times r$ . It has been recently shown that odd values of  $r$  must be chosen to avoid some possible weaknesses [13]. In most instances of these systems appeared in previous literature, the matrix  $\mathbf{H}$  is regular, although it has been shown in [7] that using irregular matrices may bring some reduction in the public key size. Thus, for the sake of comparison, we focus on regular parity-check matrices, having all the columns with weight  $d_v$  and all the rows with weight  $d_c = n_0 \cdot d_v$ . When  $d_v$  is much smaller than  $r$ , we say that the code is an LDPC code. MDPC

codes are a special class of LDPC codes, characterized by moderately small values of  $d_v$  (in the order of several tenths).

#### A. Key generation

The private key is formed by the secret parity-check matrix  $\mathbf{H}$  and two other non-singular matrices: a  $k \times k$  scrambling matrix  $\mathbf{S}$  and an  $n \times n$  transformation matrix  $\mathbf{Q}$ . The latter is defined as a sparse matrix with average row and column weight  $m \geq 1$  ( $m$  is not necessarily an integer, since  $\mathbf{Q}$  can be irregular). Both  $\mathbf{S}$  and  $\mathbf{Q}$  have quasi-cyclic (QC) form, that is, they consist of circulant sub-matrices with size  $r \times r$ . When QC-MDPC codes are used,  $\mathbf{Q}$  boils down to an  $n \times n$  permutation matrix, and we have  $m = 1$ . In this case, as done in [5], the secret permutation can even be avoided and  $\mathbf{Q}$  eliminated (i.e.,  $\mathbf{Q} = \mathbf{I}_{n \times n}$ , the  $n \times n$  identity matrix).

The public key is obtained as  $\mathbf{G}' = \mathbf{S}^{-1} \cdot \mathbf{G} \cdot \mathbf{Q}^{-1}$ , where  $\mathbf{G}$  is a systematic generator matrix obtained from  $\mathbf{H}$ . The role of  $\mathbf{S}$  is to make the public generator matrix non-systematic. However, if we consider a CCA2 secure conversion of the system [2],  $\mathbf{G}'$  can be in systematic form, therefore  $\mathbf{S}$  can be eliminated (i.e.,  $\mathbf{S} = \mathbf{I}_{k \times k}$ ). With  $\mathbf{G}'$  in systematic form, and exploiting its QC structure, the public key size is  $(n_0 - 1) \cdot r$  bits. Such a size is considerably smaller than for classical Goppa code-based instances with the same security level.

It follows from the public key definition that the public code admits a parity-check matrix in the form  $\mathbf{H}' = \mathbf{H} \cdot \mathbf{Q}^T$ , that is sparse. Since both  $\mathbf{H}$  and  $\mathbf{Q}$  are indeed very sparse,  $\mathbf{H}'$  is very likely the sparsest parity-check matrix of the public code. For this reason, it can be the target of a key recovery attack, as we will see in Section IV-C.

#### B. Encryption

In order to encrypt her message, Alice gets Bob's public key  $\mathbf{G}'$ , divides her message into  $k$ -bit vectors and, for each of them, generates a random intentional error vector  $\mathbf{e}$  with weight  $t$ . Finally, she encrypts  $\mathbf{u}$  into  $\mathbf{x}$  as follows:

$$\mathbf{x} = \mathbf{u} \cdot \mathbf{G}' \oplus \mathbf{e} = \mathbf{c}' \oplus \mathbf{e}, \quad (2)$$

where  $\oplus$  denotes modulo-2 addition. In fact, in all existing McEliece cryptosystems, independently of the family of codes used, the intentional errors are bit flipping errors. This means that  $\mathbf{e}$  is a binary vector, and the bits of the codeword  $\mathbf{c}'$  which are at positions corresponding to the support of  $\mathbf{e}$  are flipped.

#### C. Decryption

In order to perform decryption, Bob first inverts the secret transformation (if used):

$$\mathbf{x}' = \mathbf{x} \cdot \mathbf{Q} = \mathbf{u} \cdot \mathbf{S}^{-1} \cdot \mathbf{G} \oplus \mathbf{e} \cdot \mathbf{Q} = \mathbf{c} \oplus \mathbf{e} \cdot \mathbf{Q}. \quad (3)$$

This way, he gets the codeword  $\mathbf{c}$  belonging to the private code, corrupted by the error vector  $\mathbf{e}' = \mathbf{e} \cdot \mathbf{Q}$ . Due to the structure of  $\mathbf{Q}$ ,  $\mathbf{e}'$  is a binary vector with weight  $\leq t' = tm$ . When  $m = 1$ , as in the case of QC-MDPC code-based systems,  $\mathbf{Q}$  is a permutation or an identity matrix, hence  $t' = t$ . Then, Bob performs LDPC decoding to correct all the errors and easily obtains  $\mathbf{u} \cdot \mathbf{S}^{-1}$  owing to systematic encoding. The message  $\mathbf{u}$  is then recovered through multiplication by  $\mathbf{S}$ .

### III. EXPLOITING REAL-VALUED INTENTIONAL NOISE

Let us consider a new system in which we use a real-valued intentional noise vector  $\mathbf{w}$  in place of the binary intentional error vector  $\mathbf{e}$ . In other terms,  $\mathbf{w}$  is a  $1 \times n$  vector containing real-valued noise samples affecting the codeword  $\mathbf{c}'$  during encryption. This way, the ciphertext  $\mathbf{x}$  is no longer a binary vector, and becomes a real-valued vector as well.

From the practical standpoint, real numbers are always represented through finite precision variables. So, we suppose to use  $q$ -bit variables to represent the entries of  $\mathbf{c}'$ ,  $\mathbf{w}$  and  $\mathbf{x}$ . The effects of the finite precision representation of real numbers can be made negligible by choosing a suitably large value of  $q$ . Obviously, the ciphertext length is increased by a factor  $q$  with respect to the classical systems, and this may seem an important drawback. However, as we will see next, exploiting real-valued intentional noise allows to achieve significant reductions in the public key size, which is the most important drawback of McEliece-type cryptosystems. Moreover, the intentional noise vector can also be exploited to carry information, as first proposed in [14]. This means that we could encode part of the secret message into the intentional noise vector, thus reducing the ciphertext expansion. Such a possibility, however, is left for future investigation.

For the sake of simplicity, in the following we describe the main procedures of the new system by focusing on the QC-MDPC code-based variant described in Section II with CCA2 secure conversion, using  $\mathbf{S} = \mathbf{I}_{k \times k}$  and  $\mathbf{Q} = \mathbf{I}_{n \times n}$ . The extension to more general QC-LDPC and QC-MDPC code-based schemes is also left for future works.

#### A. Key generation

As in the original system, the private key is formed by an  $r \times n$  secret parity-check matrix  $\mathbf{H}$  in the form (1), from which a  $k \times n$  generator matrix  $\mathbf{G}$  is obtained, in systematic form. Since  $\mathbf{S} = \mathbf{I}_{k \times k}$  and  $\mathbf{Q} = \mathbf{I}_{n \times n}$ , the public key is  $\mathbf{G}' = \mathbf{G}$ , with size  $(n_0 - 1) \cdot r$  bits.

#### B. Encryption

There are several possibilities to extend the encryption map (2) to the case in which we use a real-valued intentional noise. Among these, we choose an encryption map that allows to exploit some LDPC coding theory concepts which are well known in the literature. In fact, a huge amount of research works have been devoted to the design and optimization of LDPC coded transmission schemes with antipodal signals (e.g., binary pulse amplitude modulation (2-PAM)) over additive white Gaussian noise (AWGN) channels. Such results can be reused in the context under consideration if decryption is performed on a vector which looks like a modulated LDPC codeword with symbols 1 in place of bits 1 and symbols  $-1$  in place of bits 0, corrupted by AWGN. For this purpose, let us consider the following encryption map

$$\mathbf{x} = 2(\mathbf{u} \cdot \mathbf{G}') - \mathbf{1} + \mathbf{w} = 2\mathbf{c}' - \mathbf{1} + \mathbf{w}, \quad (4)$$

where  $\mathbf{1}$  is the  $1 \times n$  all-one vector. The statistical properties of  $\mathbf{w}$  can obviously be fixed a priori. We suppose that  $\mathbf{w}$  is

filled with the samples of a Gaussian variable with mean 0 and standard deviation  $\sigma$ . As we will see in Section IV-B, the generation of the vector  $\mathbf{w}$  may require more than one attempt, since some results need to be discarded for security reasons.

### C. Decryption

Since  $\mathbf{G}' = \mathbf{G}$ , the ciphertext  $\mathbf{x}$  received by Bob coincides with a 2-PAM modulated version of a codeword  $\mathbf{c}' = \mathbf{c}$  belonging to the private code, corrupted by the intentional noise vector  $\mathbf{w}$ , that contains AWGN samples. As in Section II-C, Bob then performs LDPC decoding to correct all the errors and recovers  $\mathbf{u}$  owing to systematic encoding.

Differently from the original system, in this new system Bob can exploit the optimal performance of message-passing LDPC decoding algorithms working on the soft reliability values associated to the received samples. One of the best algorithms of this type is the sum-product algorithm (SPA) with log-likelihood ratios (LLRs), that we consider in the following. In order to perform decoding through the LLR-SPA, Bob needs to compute the LLR of each codeword bit, defined as

$$\Lambda(x_i) = \log \left[ \frac{p(x_i|c_i = 1)}{p(x_i|c_i = 0)} \right], \quad (5)$$

where  $x_i$  is the symbol corresponding to the codeword bit  $c_i$  impaired with noise, and  $p(x_i|c_i)$  is the probability density function (p.d.f.) of  $x_i$  conditioned on  $c_i$ . By taking into account that we have 2-PAM signals impaired with AWGN, through simple calculations (5) can be rewritten as

$$\Lambda(x_i) = \frac{2x_i}{\sigma^2}, \quad (6)$$

where  $\sigma$  is the AWGN standard deviation.

## IV. SECURITY ASSESSMENT

In this section we assess the security of the proposed cryptosystem by considering both classical attacks and newly developed attacks aimed at exploiting the real-valued intentional noise. There are two main types of attacks which may be mounted against these systems: decoding attacks (DAs) and key recovery attacks (KRAs). While the former are aimed at decrypting one or more ciphertexts without knowing the private key, the latter aim at recovering the private key from the public key. The soft reliability information about the ciphertext bits that is available in the proposed system may facilitate DAs, thus helping an attacker to decrypt an intercepted ciphertext. For this reason, we consider classical DAs, but also devise new DAs exploiting soft reliability information.

### A. Classical decoding attacks

In a DA, the adversary intercepts a ciphertext  $\mathbf{x}$  and aims at correcting all the intentional errors added during encryption. If this succeeds, he can then invert the encoding map and recover the cleartext. The most dangerous DAs against the original LDPC and MDPC code-based cryptosystems are those exploiting information set decoding (ISD) algorithms. These techniques stem from a family of probabilistic algorithms aimed at finding low weight codewords in general linear block

codes, first introduced by Leon [15] and Stern [16]. Indeed, it can be shown that finding the binary error vector  $\mathbf{e}$  that has been used in (2) to obtain the ciphertext is very similar to searching for a low weight codeword in an extended version of the public code.

These algorithms have known great advances in recent years [17]–[20]. Today, one of the best known algorithms to search for low weight codewords in a linear block code is that introduced in [20]. Its work factor in the finite code length regime has been computed in closed form in [5, Appendix B]. For a code with length  $n$  and dimension  $k$  in which a (single) codeword with weight  $w$  is searched, we define this quantity as  $WF_{\text{BJMM}}(n, k, w)$ , representing the number of elementary operations which are needed to successfully complete the algorithm execution, on average.

The QC nature of the codes we consider facilitates such a task, since each block-wise cyclically shifted version of a ciphertext is still a valid ciphertext. Therefore, an attacker could consider all the QC shifts of an intercepted ciphertext, and search for one among as many shifted versions of the error vector. For codes in the form (1), the number of possible QC shifts of a ciphertext is  $r$ , and the corresponding advantage in terms of the algorithm complexity is in the order of  $\sqrt{r}$  [21]. Therefore, the work factor of decoding attacks against the original systems can be computed as

$$WF_{\text{DA}}^{(1)}(t) = \frac{1}{\sqrt{r}} WF_{\text{BJMM}}(n, k, t), \quad (7)$$

where  $t$  is the weight of the vector  $\mathbf{e}$  used during encryption. An attack of this kind can also be mounted against the new cryptosystem by applying hard-decision, discarding the soft reliability information and trying to correct the bit errors induced by the intentional noise.

### B. Soft reliability information-aided decoding attacks

According to the proposed approach, both Bob and Eve may take advantage of the soft reliability information about each bit of any ciphertext. This facilitates Bob, which may exploit powerful iterative soft-decision decoding algorithms for LDPC and MDPC codes. However, the same information can also be exploited by Eve to mount a decoding attack.

A first attempt that Eve can make is to also use an iterative soft-decision decoding algorithm to decode the public code. The performance of these decoders is actually difficult to predict from a theoretical standpoint. However, an ultimate bound on their performance can be computed through the density evolution technique [11]. This provides the maximum noise level which can be compensated under the hypothesis of infinite length codes with absence of closed loops in their associated graphs. Obviously, when practical, finite length codes with cycles in their graphs are used, the maximum noise level which still allows error correction is bounded away from the density evolution threshold. However, the latter is still useful in our case, since it represents an ultimate limit. Therefore, if the intentional noise level used during encryption is above the density evolution threshold, we are sure that error



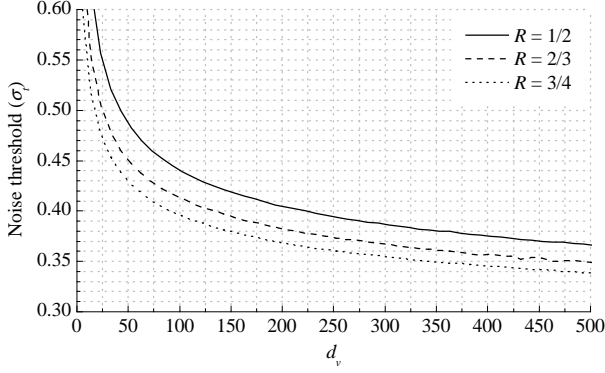


Fig. 1. Gaussian noise threshold values ( $\sigma_t$ ) found through density evolution for regular codes with parity-check matrix column weight  $d_v$  and rate  $R = 1/2, 2/3, 3/4$ .

correction cannot be performed through iterative soft-decision algorithms working on the public code, independently of the code length. The density evolution threshold decreases as long as the parity-check matrix column weight increases, as shown in Fig. 1. This is the reason why these decoding algorithms are very likely inefficient on the public code, which has a dense parity-check matrix, unless a key recovery attack is first successfully accomplished.

However, even when the intentional noise level is above the density evolution threshold and iterative soft-decision decoding algorithms are ineffective, an attacker could still exploit the soft reliability information to facilitate classical DAs. For this purpose, the attacker could proceed as follows:

- 1) sort the ciphertext bits in decreasing reliability (*i.e.*,  $|\Lambda(x_i)|$ ) order,
- 2) select the  $t_f$  least reliable ciphertext bits, suppose that they are in error and flip them,
- 3) use ISD to correct the residual bit errors.

The work factor of such an attack procedure can be computed as follows (mathematical derivations are omitted for the sake of brevity). The probability that the ciphertext bit at position  $i$  in the ordered list is in error due to an intentional Gaussian noise with mean 0 and standard deviation  $\sigma$  can be computed as

$$P_{e,i} = n \binom{n-1}{i-1} \int_0^{+\infty} g_{-1,\sigma}(x) [P(x)]^{i-1} [1 - P(x)]^{n-i} dx, \quad (8)$$

where  $g_{-1,\sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x+1)^2}{2\sigma^2}}$  is the p.d.f. of a Gaussian variable with mean  $-1$  and standard deviation  $\sigma$  and

$$P(x) = 1 - \frac{1}{2} \left[ \operatorname{erfc} \left( -\frac{x-1}{\sigma\sqrt{2}} \right) - \operatorname{erfc} \left( \frac{x+1}{\sigma\sqrt{2}} \right) \right]. \quad (9)$$

The probability that all the  $t_f$  bits flipped by the attacker are indeed in error can then be computed as

$$P_f = \prod_{i=0}^{t_f-1} P_{e,n-i}. \quad (10)$$

The overall work factor of the decoding attack aided by the soft reliability information can be obtained as

$$WF_{DA}^{(2)} = \frac{1}{\sqrt{r}} \frac{WF_{BJMM}(n, k, t^* - t_f)}{P_f}, \quad (11)$$

where  $t^*$  is the total number of bit errors on the ciphertext induced by the intentional Gaussian noise. Obviously, an attacker is free to choose the value of  $t_f$  that minimizes the work factor expressed by (11).

If we use a purely random noise, the value of  $t^*$  follows a binomial distribution with mean  $\hat{t} = \frac{n}{2} \operatorname{erfc} \left( \frac{1}{\sigma\sqrt{2}} \right)$ . Therefore, some ciphertexts may experience small values of  $t^*$ , and hence may be more vulnerable to attacks of this type. To avoid this risk, we can require Alice to discard those vectors  $\mathbf{w}$  corresponding to values of  $t^*$  falling below some threshold  $\underline{t}$ , and compute the work factor considering the worst case in which  $t^* = \underline{t}$ . We have verified through numerical simulations that a simple and effective choice is to impose that  $t^* \geq \underline{t} = \hat{t}$ . This does not require Alice to perform too many attempts to generate the vector  $\mathbf{w}$  (half of them are successful on average) while allowing to achieve good security levels with compact keys.

### C. Key recovery attacks

KRAs are aimed at recovering the private key from the public key. Even when the private key is not exactly recovered, the attack may be successful by finding an alternative private key which is still useful for an attacker to perform decoding. An efficient way to recover a sparse parity-check matrix for the public code ( $\mathbf{H}'$ ) is to search for its rows in the dual of the public code. When  $\mathbf{H}'$  is successfully recovered, it can then be used by an attacker to recover  $\mathbf{H}$  by exploiting its sparsity, or to perform LDPC decoding and correct the intentional errors. In general, the matrix  $\mathbf{H}'$  has column weight  $d'_v = m \cdot d_v$  (that is,  $d'_v = d_v$  for the QC-MDPC code-based system we consider) and row weight  $d'_c = n_0 \cdot d'_v$ . Therefore,  $d'_v$  must be large enough to make finding the rows of  $\mathbf{H}'$  in the dual of the public code computationally infeasible for an attacker.

Since the codes are QC with parity-check matrices in the form (1), all the rows of  $\mathbf{H}'$  are obtained as the QC shifts of one of them. This reduces the attack complexity by a factor equal to the number of rows of  $\mathbf{H}'$ , *i.e.*,  $r$ . Therefore, the work factor of a KRA can be computed as

$$WF_{KRA} = \frac{1}{r} WF_{BJMM}(n, r, d'_c). \quad (12)$$

### V. EXAMPLES

Let us focus on 80-bit security. The McEliece cryptosystem with the shortest public key size known in the literature is reported in [5], and achieves 4801-bit public keys (with CCA2-security conversion) using codes with length  $n = 9602$ , rate  $1/2$  and a decoding failure rate (DFR) in the order of  $10^{-7}$  or less. Such a system uses  $t = 84$  binary intentional errors and QC-MDPC codes with public and private parity-check matrix column weight  $d'_v = d_v = 45$ .

Let us consider a similar system with the same value of  $d'_v = d_v$ , and suppose that the same number of bit errors are induced by an intentional Gaussian noise, i.e.,  $\underline{t} = \hat{t} = 84$ . If we reduce the code length to  $n = 7202$ , we obtain  $\sigma = \frac{1}{\sqrt{2}\text{erfc}^{-1}(2\hat{t}/n)} = 0.44091$ . We have verified through numerical simulations that, for these values of  $n$  and  $\sigma$ , the LLR-SPA decoder is still able to compensate the intentional noise with a DFR in the order of  $10^{-7}$ .

Concerning DAs and KRAs, their work factors are  $WF_{DA}^{(2)} = 2^{80.49}$  (minimum for  $t_f = 0$ ) and  $WF_{KRA} = 2^{80.17}$ , respectively. According to Fig. 1, the density evolution threshold for codes with rate  $1/2$  falls below  $0.4$  for parity-check matrix column weights  $> 223$ . Unless a KRA is performed, the parity-check matrices of the public code which are available to an attacker are dense, with column weight in the order of one thousand or more. Hence, iterative soft-decision decoders cannot be used to cancel the intentional noise. Therefore, this system is able to achieve 80-bit security with 3601-bit public keys, that is, about 25% less than the best known solution.

If we focus on 128-bit security, the solution provided in [5] that achieves the smallest public key is that using codes with rate  $1/2$ ,  $n = 19714$ ,  $d'_v = d_v = 71$  and  $t = 134$  intentional errors. Considering  $n = 15770$  and an intentional Gaussian noise with  $\underline{t} = \hat{t} = 134$  yields  $\sigma = 0.41897$ , that is still above the density evolution threshold for dense matrices. We have verified through simulations that a QC-MDPC code with rate  $1/2$ ,  $n = 15770$  and  $d_v = 71$  is able to compensate such a noise level under LLR-SPA decoding, with a DFR in the order of  $10^{-7}$ . In this case, the attack work factors are  $WF_{DA}^{(2)} = 2^{129.06}$  (minimum for  $t_f = 0$ ) and  $WF_{KRA} = 2^{130.24}$ . Therefore, 128-bit security can be achieved with 7885-bit public keys, that is, about 20% less than the best known solution (reported in [5] and requiring 9857-bit public keys).

In both these cases, the smallest work factors are found when no bits are flipped by the attacker based on their reliabilities (i.e.,  $t_f = 0$ ). However, this situation changes when the code rate is larger than  $1/2$ , since the advantage an attacker gains by exploiting flipped bits becomes significant. For example, if we consider a code with  $n = 10779$  and rate  $2/3$ , with  $\underline{t} = \hat{t} = 53$  (that implies  $\sigma = 0.38736$ ) and  $t_f = 0$ , we obtain  $WF_{DA}^{(2)} = 2^{80.84}$ . Instead, if the attacker tries to flip the least reliable bits before performing ISD, the attack work factor can be reduced to  $WF_{DA}^{(2)} = 2^{75.53}$  (minimum for  $t_f = 22$ ) and the value of  $\underline{t}$  must be consequently increased in order to achieve 80-bit security.

## VI. CONCLUSION

We have verified that using real-valued intentional noise during encryption can be highly beneficial in order to achieve QC-MDPC code-based cryptosystems with compact keys. Our results show that public key size reductions up to 25% with respect to the best known solutions can be obtained. Future investigations will involve the chance to jointly use binary and real-valued intentional errors, as well as real-valued noise samples conveying part of the secret message.

## REFERENCES

- [1] R. J. McEliece, "A public-key cryptosystem based on algebraic coding theory," *DSN Progress Report*, pp. 114–116, 1978.
- [2] D. J. Bernstein, T. Lange, and C. Peters, "Attacking and defending the McEliece cryptosystem," in *Post-Quantum Cryptography*, ser. Lecture Notes in Computer Science. Springer Verlag, 2008, vol. 5299, pp. 31–46.
- [3] M. Baldi, M. Bodrato, and F. Chiaraluce, "A new analysis of the McEliece cryptosystem based on QC-LDPC codes," in *Security and Cryptography for Networks*, ser. Lecture Notes in Computer Science. Springer Verlag, 2008, vol. 5229, pp. 246–262.
- [4] M. Baldi, M. Bianchi, and F. Chiaraluce, "Security and complexity of the McEliece cryptosystem based on QC-LDPC codes," *IET Information Security*, vol. 7, no. 3, pp. 212–220, Sep. 2012.
- [5] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. S. L. M. Barreto. (2012) MDPC-McEliece: New McEliece variants from moderate density parity-check codes. [Online]. Available: <http://eprint.iacr.org/2012/409>
- [6] M. Baldi, M. Bianchi, and F. Chiaraluce, "Optimization of the parity-check matrix density in QC-LDPC code-based McEliece cryptosystems," in *Proc. IEEE ICC 2013 - Workshop on Information Security over Noisy and Lossy Communication Systems*, Budapest, Hungary, Jun. 2013.
- [7] M. Baldi, M. Bianchi, N. Maturo, and F. Chiaraluce, "Improving the efficiency of the LDPC code-based McEliece cryptosystem through irregular codes," in *Proc. IEEE Symposium on Computers and Communications (ISCC 2013)*, Split, Croatia, Jul. 2013.
- [8] F. P. Biasi, P. S. L. M. Barreto, R. Misoczki, and W. V. Ruggiero, "Scaling efficient code-based cryptosystems for embedded platforms," *Journal of Cryptographic Engineering*, vol. 4, no. 2, pp. 123–134, Jun. 2014.
- [9] I. Von Maurich, T. Oder, and T. Güneysu, "Implementing QC-MDPC McEliece encryption," *ACM Transactions on Embedded Computing Systems*, vol. 14, no. 3, pp. 44:1–44:27, May 2015.
- [10] R. G. Gallager, *Low-density parity-check codes*. M.I.T. Press, 1963.
- [11] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
- [12] M. Baldi, F. Bambozzi, and F. Chiaraluce, "On a family of circulant matrices for quasi-cyclic low-density generator matrix codes," *IEEE Trans. Inform. Theory*, vol. 57, no. 9, pp. 6052–6067, Sep. 2011.
- [13] C. Löndahl, T. Johansson, M. Koochak Shooshtari, M. Ahmadian-Attari, and M. Aref, "Squaring attacks on McEliece public-key cryptosystems using quasi-cyclic codes of even dimension," *Designs, Codes, and Cryptography*, Jun. 2015, published online.
- [14] J. Riek, "Observations on the application of error correcting codes to public key encryption," in *Proc. IEEE International Carnahan Conference on Security Technology: Crime Countermeasures*, Lexington, KY, USA, Oct. 1990, pp. 15–18.
- [15] J. Leon, "A probabilistic algorithm for computing minimum weights of large error-correcting codes," *IEEE Trans. Inform. Theory*, vol. 34, no. 5, pp. 1354–1359, Sep. 1988.
- [16] J. Stern, "A method for finding codewords of small weight," in *Coding Theory and Applications*, ser. Lecture Notes in Computer Science, G. Cohen and J. Wolfmann, Eds. Springer Verlag, 1989, vol. 388, pp. 106–113.
- [17] C. Peters, "Information-set decoding for linear codes over  $F_q$ ," in *Post-Quantum Cryptography*, ser. Lecture Notes in Computer Science. Springer Verlag, 2010, vol. 6061, pp. 81–94.
- [18] A. May, A. Meurer, and E. Thomae, "Decoding random linear codes in  $O(2^{0.054n})$ ," in *ASIACRYPT 2011*, ser. Lecture Notes in Computer Science. Springer Verlag, 2011, vol. 7073, pp. 107–124.
- [19] D. J. Bernstein, T. Lange, and C. Peters, "Smaller decoding exponents: ball-collision decoding," in *CRYPTO 2011*, ser. Lecture Notes in Computer Science. Springer Verlag, 2011, vol. 6841, pp. 743–760.
- [20] A. Becker, A. Joux, A. May, and A. Meurer, "Decoding random binary linear codes in  $2^{n/20}$ : How  $1 + 1 = 0$  improves information set decoding," in *Advances in Cryptology - EUROCRYPT 2012*, ser. Lecture Notes in Computer Science, D. Pointcheval and T. Johansson, Eds. Springer Verlag, 2012, vol. 7237, pp. 520–536.
- [21] N. Sendrier, "Decoding one out of many," in *Post-Quantum Cryptography*, ser. Lecture Notes in Computer Science, B.-Y. Yang, Ed. Springer Verlag, 2011, vol. 7071, pp. 51–67.